UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,166	12/05/2003	Manish Anand Bhide	JP920030164US1	8579

7590 06/25/2007

Frederick W. Gibb, III
McGinn & Gibb, PLLC
Suite 304
2568-A Riva Road
Annapolis, MD 21401

EXAMINER	
SYED, FARHAN M	

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
06/25/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/729,166
Filing Date: December 05, 2003
Appellant(s): BHIDE ET AL.

Mohammad S. Rahman (Reg. No. 43,029)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 28 February 2007 appealing from the Office action mailed 06 October 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Dayal, Umeshwar, et al., "Active Database Systems," Addison-Wesley, September 2004, pp 0-24

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-15 are rejected under 35 U.S.C. 102(b) as being anticipated by a non-patent literature titled "Active Database Systems" by Umeshwar Dayal, Eric N. Hanson, and Jennifer Wisdom, Addison-Wesley, Sept. 1994 (and known hereinafter as Dayal).

As per claims 1 and 13-15, Dayal teaches a method of monitoring events in a database, said method comprising: (i.e. *"An active database system, in contrast, is a database system that monitors situations of interest, and when they occur, triggers an appropriate response in a timely manner."* The preceding text clearly indicates that monitoring events in a database is an active database system.)(Page 1, paragraph 3): storing in said database at least one database rule (i.e. *"The desired behavior is expressed in production rules (also called event-condition-action rules), which are defined and stored in the database."* The preceding text clearly indicates that storing production rules indicates that at least one rule is stored in the database.)(Page 1, paragraph 3); mapping temporal constraints of an event of the database rule to corresponding temporal events (i.e. *"An inference engine cycles through all the rules in the system, matching the condition parts of the* *rules with the data in working memory."* The preceding text clearly indicates that mapping is matching and temporal constraints are conditions that contain time and event.)(Page 1, paragraph 3); changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of the database rule (i.e. *"The action part may modify the working memory, possibly according to the matched data, and the cycle continues until no more rules match."* The preceding text clearly indicates that changing the temporal constraints is modifying the condition, which is the action part of the working memory.)(Page 1, paragraph 3); registering alarms associated with a start and end of a lifespan of each temporal event (i.e. *"...a rule is triggered whenever one or more of its triggering operation occurs." "In addition, active database systems must provide mechanisms for event detection and rule triggering, for condition testing, for rule action execution, and for user development of rule applications."* The preceding text clearly indicates that registering alarms associated with each temporal event is an illustration of a triggering operation. Furthermore, using an active database, an ordinary person skilled in the art may reasonably infer that such alarms are anticipated in the database system, because for an active database system to be active, it must do event detection, and registering alarms illustrates such a point.)(Page 14, section 3.3; Pages 17-18; section 4); selectively deploying and selectively permanently removing the temporal events from database based upon said changed temporal constraints (i.e. *"Rules are structured objects, having events, conditions and actions as their components. Like any object, rules can be created, deleted, or modified. In addition, rule objects have some special operations, including: fire, which causes a rule to be triggered; enable, which cases a rule to be activated; disable, which causes a rule to be deactivated (so that it won't be triggered even if its triggered event occurs." "Rules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative."* The preceding text clearly indicates that selectively deploying is enable, which cases the trigger to be activated and selectively removing is disable, which cases the trigger to be deactivated. Furthermore when a table is dropped, an ordinary person skilled in the art anticipates that the table is either deleted or removed from the database, thus it can be inferred that the table is permanently removed; which by proxy would infer the temporal events would too be permanently removed.)(Page 3, paragraph 3; Page 9, section 2.5); upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarms associated with the permanent removed temporal event (i.e. *"Rules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative."* When a table is dropped, an ordinary person skilled in the art may reasonably infer that the skilled person can remove the table from the database and thereby manually remove all temporal constraints and alarms associated with the targeted table. The manual removal of the temporal constraints and associated alarms are consistent with a natural human phenomenon, when the skilled person performs duly maintenance on the database. Whether manually or dynamically performed, no novelty exists in optimizing system efficiency.)(Page 9, section 2.5).

As per claim 2, Dayal teaches a method further comprising removing from the database temporal events that cannot evaluate as true (i.e. *"Most database rule systems handle errors during rule processing by aborting the current transaction, since this is how conventional database systems typically handle errors during transaction processing."*)(Page 17, paragraph 2).

As per claim 3, Dayal teaches a method further comprising limiting the lifespan of an event to the overlapping period (i.e. spawning)(page 17, paragraph 3) of the lifespan of a parent event (i.e. *"The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the* *firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."*)(Page 17, paragraph 3).

As per claim 4, Dayal teaches a method further comprising changing the lifespan of an event to omit periods in which the event cannot evaluate as true (i.e. *"In general, when a triggered rule is executed in Ariel, the rule processes the entire set of triggering changes, including both the user-generated changes that initiated the rule processing and any subsequent changes made by rule actions" "Some languages have sequential execution semantics, while others allow concurrent execution. With either sequential or concurrent execution semantics, there is also the issue of whether one rule can trigger the execution of another rule or of (another instance of) the same rule. Clearly, if such nested triggering is allowed, termination is a concern."* The preceding text clearly indicates that when the author mentioned termination is a concern, an ordinary person skilled in the art understands that termination or abort must take place when the event cannot evaluate as true.)(Page 12, paragraph 1; page 11, paragraph 1).

As per claim 5, Dayal teaches a method further comprising assigning a lifespan of an event having an undefined lifespan as the lifespan of a parent event (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." "The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."*)(Page 3, paragraph 5; page 17, paragraph 3).

As per claim 6, Dayal teaches a method further comprising propagating the lifespan or context of the parent node to all children nodes of the parent node (i.e. *"The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."*)(Page 17, paragraph 3).

As per claim 7, Dayal teaches a method wherein a lifespan of an event is expressed as a predetermined duration of time (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." "Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)."*)(Page 3, paragraph 5; page 5, paragraph 2).

As per claim 8, Dayal teaches a method wherein the lifespan is dependent upon the associated event (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action."*)(Page 3, paragraph 5).

As per claim 9, Dayal teaches a method wherein the lifespan ends at a predetermined time, or recurs at a predetermined period of time (i.e. *"Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)."*)(Page 5, paragraph 2).

As per claim 10 and 18, Dayal teaches a method further comprising combining events using a sequence operator to form a composite event having a time span (i.e. *"Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)." "When an instance of this event type occurs, the formal parameters are bound to a specific employee (the one whose salary is being updated) and two specific integers (this employee's old salary and new salary)."*)(Page 5, paragraph 2; page 6, paragraph 3).

As per claim 11, Dayal teaches a method further comprising associating a lifespan with the sequence operator (i.e. *"Most importantly, unlike in AI systems, in active database systems rule processing is integrated with conventional database activity – queries, modifications, and transactions – and it is this activity that causes rules to become triggered and initiates rule processing."* The preceding text clearly indicates that a lifespan is contained within a rule that processes queries, modifications, and transactions, and the sequence operator is the active database system that initiates rule processing.)(Page 10, paragraph 3).

As per claim 12, Dayal teaches a method further comprising the step of storing a database rule as an event-condition-action (ECA) rule (i.e. *"The desired behavior is expressed in production rules (also called event-condition-action rules), which are defined and stored in the database."* The preceding text clearly indicates that ECA rules are stored in a database.)(Page 1, paragraph 3).

As per claims 16 and 19, Dayal teaches a method further comprises using a separate device external to said database to detect the combined events (i.e. *"The* *implementation of an active database system can include many useful features that support the rule programmer. Features for analyzing rule processing include the ability to trace rule execution, to display the current set of triggered rules, to query and browse the set of rules, and to cross-reference rules and data. Other useful features include the ability to control errors in rule programs, to activate and deactivate selected rules or groups of rules while the database system is processing transactions, and to experiment with rules on an off-line subset of a working database."* The preceding text clearly indicates that a separate device external to said database is an example of one of many useful features that support the rule programmer.)(Page 19, section 4.2).

As per claims 17 and 20, Dayal teaches a method wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database (i.e. *"When the triggering event occurs, the condition is evaluated against the database; if the condition is satisfied, the action is executed."*)(Page 2, section 1).

(10) Response to Argument

The Applicant argues Dayal does not teach or suggest "registering alarms associated with a start and end of a lifespan of each temporal event; selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and upon reaching the said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event."

The Examiner disagrees. Dayal teaches registering (i.e. creating)(section 2, paragraph 3) alarms (i.e. rules... *"Rules are structured objects, having events, conditions, and actions as their components."*)(section 2, page 3) associated with a start and end of a lifespan (i.e.

conditions)(section 2, page 3) of each temporal event (i.e. triggering event)(section 2, page 3) (i.e. *"...a rule is triggered whenever one or more of its triggering operation occurs." "In addition, active database systems must provide mechanisms for event detection and rule triggering, for condition testing, for rule action execution, and for user development of rule applications."* The preceding text clearly indicates that registering alarms associated with each temporal event is an illustration of a triggering operation. Furthermore, using an active database, an ordinary person skilled in the art may reasonably infer that such alarms are anticipated in the database system, because for an active database system to be active, it must do event detection, and registering alarms illustrates such a point.)(Page 14, section 3.3; Pages 17-18; section 4); selectively deploying (i.e. enabling rules)(section 2, page 3) and selectively permanently removing (i.e. deleted..."Like any object, rules can be created, **deleted**, or modified.")(Section 2, page 3) the temporal events from database based upon said changed temporal constraints (i.e. *"Rules are structured objects, having events, conditions and actions as their components. Like any object, rules can be created, deleted, or modified. In addition, rule objects have some special operations, including: fire, which causes a rule to be triggered; enable, which cases a rule to be activated; disable, which causes a rule to be deactivated (so that it won't be triggered even if its triggered event occurs." "Rules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative."* The preceding text clearly indicates that selectively deploying is enable, which cases the trigger to be activated and selectively removing is disable, which cases the trigger to be deactivated. Furthermore when a table is dropped, an ordinary person skilled in the art anticipates that the table is either deleted or removed from the database, thus it can be inferred that the table is permanently removed; which by proxy would infer the temporal events would too be permanently removed.)(Page 3, paragraph 3; Page 9, section 2.5); upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarms associated with the permanent removed temporal event (i.e. *"Rules refer to particular tables, and so are subject* *to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative."* When a table is dropped, an ordinary person skilled in the art may reasonably infer that the skilled person can remove the table from the database and thereby manually remove all temporal constraints and alarms associated with the targeted table. The manual removal of the temporal constraints and associated alarms are consistent with a natural human phenomenon, when the skilled person performs duly maintenance on the database. Whether manually or dynamically performed, no novelty exists in optimizing system efficiency.)(Page 9, section 2.5).

Applicant argues "These features are neither taught or suggested in Dayal because in Dayal the temporal events are not physically completely removed from the database."

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., **physically completely removed** from the database) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

As per claim 2, Applicant argues "Dayal fails to teach each and every feature of the Applicant's claimed invention."

The Examiner disagrees. Dayal teaches a method further comprising removing (i.e. delete)(section 2) from the database temporal events (i.e. triggering event)(section2) that cannot evaluate as true (i.e. *"Most database rule systems handle errors during rule processing by* *aborting the current transaction, since this is how conventional database systems typically handle errors during transaction processing."*)(Page 17, paragraph 2).

As per claim 3, Applicant argues "limiting the lifespan of an event to the overlapping period of the lifespan of the parent event" is not taught by the prior art of record.

The Examiner disagrees. Dayal teaches a method further comprising limiting the lifespan of an event to the overlapping period (i.e. spawning)(page 17, paragraph 3) of the lifespan of a parent event (i.e. *"The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."* The preceding text clearly indicates that spawning allows an overlapping period for an event to occur, to which the failure of the event (i.e. the event is not triggered) is returned back to the parent. That is the prior art of record clearly suggests that a trigger event (i.e. child event) may be contained inside another trigger event (i.e. parent event).)(Page 17, paragraph 3).

As per claim 4, Applicant argues Dayal does not teach "changing the lifespan of an event to omit period in which the event cannot evaluate as true".

The Examiner disagrees. Dayal teaches a method further comprising changing the lifespan of an event to omit periods in which the event cannot evaluate as true (i.e. *"In general, when a triggered rule is executed in Ariel, the rule processes the entire set of triggering changes, including both the user-generated changes that initiated the rule processing and any* *subsequent changes made by rule actions" "Some languages have sequential execution semantics, while others allow concurrent execution. With either sequential or concurrent execution semantics, there is also the issue of whether one rule can trigger the execution of another rule or of (another instance of) the same rule. Clearly, if such nested triggering is allowed, termination is a concern."* The preceding text clearly indicates that when the author mentioned termination is a concern, an ordinary person skilled in the art understands that termination or abort must take place when the event cannot evaluate as true.)(Page 12, paragraph 1; page 11, paragraph 1).

As per claim 5, Applicant argues Dayal does not teach the limitations as recited in claim 5.

The Examiner disagrees. Dayal teaches a method further comprising assigning a lifespan of an event having an undefined lifespan as the lifespan of a parent event (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." "The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."*)(Page 3, paragraph 5; page 17, paragraph 3).

As per claim 6, the Applicant argues Dayal does not teach the limitations as recited in claim 6.

The Examiner disagrees. Dayal teaches a method further comprising propagating the lifespan or context of the parent node to all children nodes of the parent node (i.e. *"The nested transaction model used in HiPAC allows some of these possibilities. When a rule* *execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction."*)(Page 17, paragraph 3).

As per claim 7, the Applicant argues Dayal does not teach the limitations as recited in claim 7.

The Examiner disagrees. Dayal teaches a method wherein a lifespan of an event is expressed as a predetermined duration of time (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." "Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)."*)(Page 3, paragraph 5; page 5, paragraph 2).

As per claim 8, the Applicant argues Dayal does not teach the limitations as recited in claim 8.

The Examiner disagrees. Dayal teaches a method wherein the lifespan is dependent upon the associated event (i.e. *"In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action."*)(Page 3, paragraph 5).

As per claim 9, the Applicant argues Dayal does not teach the limitations as recited in claim 9.

The Examiner disagrees. Dayal teaches a method wherein the lifespan ends at a predetermined time, or recurs at a predetermined period of time (i.e. *"Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)."*)(Page 5, paragraph 2).

As per claim 10 and 18, the Applicant argues Dayal does not teach the limitations as recited in claims 10 and 18.

The Examiner disagrees. Dayal teaches a method further comprising combining events using a sequence operator to form a composite event having a time span (i.e. *"Some languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)." "When an instance of this event type occurs, the formal parameters are bound to a specific employee (the one whose salary is being updated) and two specific integers (this employee's old salary and new salary)."*)(Page 5, paragraph 2; page 6, paragraph 3).

As per claim 11, the Applicant argues Dayal does not teach the limitations as recited in claim 11.

The Examiner disagrees. Dayal teaches a method further comprising associating a lifespan with the sequence operator (i.e. *"Most importantly, unlike in AI systems, in active database systems rule processing is integrated with conventional database activity – queries, modifications, and transactions – and it is this activity that causes rules to become triggered and initiates rule processing."* The preceding text clearly indicates that a lifespan is contained within a rule that processes queries, modifications, and transactions, and the sequence operator is the active database system that initiates rule processing.)(Page 10, paragraph 3).

As per claim 12, the Applicant argues Dayal does not teach the limitations as recited in claim 12.

The Examiner disagrees. Dayal teaches a method further comprising the step of storing a database rule as an event-condition-action (ECA) rule (i.e. *"The desired behavior is expressed in production rules (also called event-condition-action rules), which are defined and stored in the database."* The preceding text clearly indicates that ECA rules are stored in a database.)(Page 1, paragraph 3).

As per claims 16 and 19, the Applicant argues Dayal does not teach the limitations as recited in claim s16 and 19.

The Examiner disagrees. Dayal teaches a method further comprises using a separate device external to said database to detect the combined events (i.e. *"The implementation of an active database system can include many useful features that support the rule programmer. Features for analyzing rule processing include the ability to trace rule execution, to display the current set of triggered rules, to query and browse the set of rules, and to cross-reference rules and data. Other useful features include the ability to control errors in rule programs, to activate and deactivate selected rules or groups of rules while the database system is processing transactions, and to experiment with rules on an off-line subset of a working database."* The preceding text clearly indicates that a separate device external to said database is an example of one of many useful features that support the rule programmer.)(Page 19, section 4.2).

As per claims 17 and 20, the Applicant argues Dayal does not teach the limitations as recited in claims 17 and 20.

The Examiner disagrees. Dayal teaches a method wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database (i.e. *"When the triggering event occurs, the condition is evaluated against the database; if the condition is satisfied, the action is executed."*)(Page 2, section 1).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Farhan Syed

Conferees:

Jeffrey Gaffin (SPE, AU 2165)

JEFFREY GAFFIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

C. Rones

CHARLES RONES
SUPERVISORY PATENT EXAMINER

Charles Rones (SPE, AU 2164)